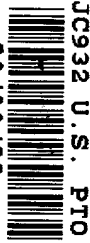


09/29/00



JC932 U.S. PTO

FISH & RICHARDSON P.C.

10-02 -00

A

4350 La Jolla Village Drive  
Suite 500  
San Diego, California  
92122

Telephone  
858 678-5070

Facsimile  
858 678-5099

Web Site  
www.fr.com

September 29, 2000

Attorney Docket No.: 10559-165001/P8249

**Box Patent Application**  
Commissioner for Patents  
Washington, DC 20231

Presented for filing is a new original patent application of:

Applicant: BRYAN R. WHITE

Title: SHARED TRANSLATION ADDRESS CACHING

Enclosed are the following papers, including those required to receive a filing date under 37 CFR §1.53(b):

	<u>Pages</u>
Specification	10
Claims	6
Abstract	1
Declaration [To be Filed at a Later Date]	
Drawing(s)	4

Enclosures:  
— Postcard.

Basic filing fee	\$690
Total claims in excess of 20 times \$18	\$0
Independent claims in excess of 3 times \$78	\$0
Fee for multiple dependent claims	\$0
Total filing fee:	\$690

CERTIFICATE OF MAILING BY EXPRESS MAIL

Express Mail Label No. EE 647 188 115 US

I hereby certify under 37 CFR §1.10 that this correspondence is being deposited with the United States Postal Service as Express Mail Post Office to Addressee with sufficient postage on the date indicated below and is addressed to the Commissioner for Patents, Washington, DC 20231

Date of Deposit September 29, 2000

Signature Francisco Robles

Typed or Printed Name of Person Signing Certificate  
Francisco Robles



BOSTON

DALLAS

DELAWARE

NEW YORK

SAN DIEGO

SILICON VALLEY

TWIN CITIES

WASHINGTON, DC



JC932 U.S. PTO  
09/676844

09/29/00

FISH & RICHARDSON P.C.

Commissioner for Patents

September 29, 2000

Page 2

No filing fee is being paid at this time. Please apply any other required fees **EXCEPT FOR THE FILING FEE**, to deposit account 06-1050, referencing the attorney document number shown above. A duplicate copy of this transmittal letter is attached.

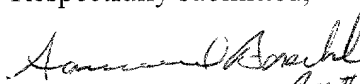
If this application is found to be incomplete, or if a telephone conference would otherwise be helpful, please call the undersigned at (212) 765-5070.

Kindly acknowledge receipt of this application by returning the enclosed postcard.

Please send all correspondence to:

SCOTT C. HARRIS  
Fish & Richardson P.C.  
45 Rockefeller Plaza, Suite 2800  
New York, NY 10111

Respectfully submitted,

  
Reg # 38,388

Scott C. Harris  
Reg. No. 32,030  
Enclosures  
SCH/dlm  
30028725 doc

# APPLICATION FOR UNITED STATES PATENT

in the name of

**Bryan R. White**

for

**Shared Translation Address Caching**

006260 44892960

Fish & Richardson P.C.  
4350 La Jolla Village Drive, Suite 500  
San Diego, CA 92122

**ATTORNEY DOCKET:**

**10559-165001 / P8249**

**DATE OF DEPOSIT:**

**EXPRESS MAIL NO.:**

*September 29, 2000*  
~~EM~~ *647 188 115* US  
*EE*

## SHARED TRANSLATION ADDRESS CACHING

### BACKGROUND

The invention relates to caching in memory controller hubs.

Microcomputer systems generally include one or more  
5 memory controller hubs that control and coordinate the  
transfer of data between the computer's system memory, central  
processing unit (CPU), and peripheral devices. Graphics  
applications may be supported by peripheral devices known as  
graphics controllers that require a memory controller hub to  
10 transfer data between the graphics controller, the system  
memory, and the CPU.

A design concern associated with microcomputer systems is  
the quality of two-dimensional (2D), three-dimensional (3D),  
and video image (collectively referred to below as "graphics")  
15 processing. High-performance graphics processing requires  
processor-intensive calculations and the fast manipulation of  
large quantities of data. Several designs have been  
implemented to achieve high-performance graphics processing  
while also reducing the cost of the complete system and  
20 allowing for upgrades to the computer system's capability.

A computer system may include a graphics controller  
coupled to local memory for storing graphics data, so that the  
amount of data that must be transferred between the graphics

controller and the system memory and/or the CPU is reduced.

Increasing the amount of local memory available to the graphics controller improves graphics performance, but also increases the cost of the computer system, because local

5 graphics memory is relatively expensive. Less local memory is required to achieve the same graphics performance, however, if a dedicated bus, e.g., an Accelerated Graphics Port (AGP), is used to couple the controller to the memory controller hub.

10 An AGP allows the controller to treat portions of system memory as dedicated local graphics memory, which reduces the amount of local memory required and lowers overall system costs.

Computer system costs also may be reduced by eliminating the peripheral graphics controller and integrating its  
15 functionality into the memory controller hub. In such a configuration the memory controller hub is better described as a graphics/memory controller hub, since it performs graphics processing functions in addition to memory control and transfer functions. Additionally, it includes one or more  
20 output ports to send graphics signals to external devices, such as cathode ray tubes (CRTs) and flat-panel monitors. A graphics/memory controller hub may be coupled to local memory for storing graphics data.

## BRIEF DESCRIPTION OF DRAWINGS

Figure 1 is a schematic block diagram of a computer system.

Figure 2 is a schematic block diagram of a graphics memory controller hub.

Figure 3 is a schematic block diagram of accelerated graphics port (AGP) functionality of a graphics memory controller hub.

## DETAILED DESCRIPTION

In a computer system, a memory controller hub may be integrated with an internal graphics controller and may interface with an external graphics device through an AGP port. Because the memory controller hub controls both graphics and memory functions it is referred to as a graphics/memory controller hub (GMCH). The GMCH provides both internal graphics processing and scalable graphics performance through an AGP interface.

The GMCH may be used in one of two mutually exclusive modes: AGP mode, in which case the GMCH uses its capability to interface with an external graphics controller and its internal graphics functionality is disabled; or Gfx mode, in which case the GMCH uses its internal graphics capability, and its ability to interface with an external graphics controller is disabled. In Gfx mode the GMCH can still interface with a

local memory module through the AGP port to provide additional graphics memory for use by the internal graphics. Whether the GMCH operates in AGP mode or Gfx mode can be determined automatically and set during the start-up sequence of the computer.

Figure 1 illustrates an exemplary computer system 1 in which the GMCH can be implemented. The computer system 1 includes a microprocessor (for example, CPU) 2 coupled to a GMCH 3, which contains a system memory controller hub. GMCH 3 may also be referred to as a "chipset" or "core logic." GMCH 3 provides an interface between CPU 2 and system memory 4, and between CPU 2 and a bus, for example, a peripheral component interconnect (PCI) or Hublink™ bus 5. Various input/output (I/O) devices 6 are coupled to PCI bus 5, which is coupled to GMCH 3 via input/output controller hub (ICH) 11. Computer system 1 may also include a graphics device 7, which may be a graphics controller coupled to local memory 8, or which may be an AGP Inline Memory Module (AIMM) that provides external local memory for the internal graphics functionality of GMCH 3. A shared AGP/local memory interface 9 provides a dedicated interface bus between GMCH 3 and graphics device 7. Graphics and video signals may be sent to a display device 10 from graphics device 7 if one is present in the computer system, or

may be sent to display device 10 from GMCH 3 if graphics device 7 is absent.

Figure 2 illustrates other details of GMCH 3, including a CPU interface 20 coupled to an AGP interface 21, a local memory interface 22, an input/output (I/O) hub interface 23, and a system memory interface 24. Graphics functions can be performed by internal graphics components 25, which include a data stream and dispatch controller 26 to manage the flow of data and various graphics engines 27 to perform graphics operations on data.

Referring to Figure 3, AGP transactions are run in a split transaction fashion in which a request for data transfer to or from system memory 4 is disconnected in time from the data transfer itself. An AGP compliant graphics device (bus master) 7a initiates a transaction with an access request. The AGP interface 21 responds to the request by directing the corresponding data transfer at a later time, which permits the AGP graphics device 7a to pipeline several access requests while waiting for data transfers to occur. As a result of pipelining, several read and/or write access requests may be simultaneously outstanding in request queues 100. Access requests can either be pipelined across an address/data bus (AD bus) 105, 107 of AGP 9 or transferred through sideband address lines 107 of AGP 9 and received by request queue 100.



Scheduler 102 processes the access requests in request queue 100. Read data are obtained from system memory 4 and are returned at the initiative of scheduler 102 through read data return queue 104 and across AD bus 105 of the AGP 9.

5 Write data are provided by AGP compliant graphics controller 7 at the direction of scheduler 102 when space is available in the write data queue 108. Thus, AGP transactions generally include interleaved access requests and data transfers.

Graphics data may be stored in system memory 4 when GMCH 10 3 operates in AGP mode in conjunction with an external AGP compliant graphics controller 7a, or when GMCH 3 operates in Gfx mode using its internal graphics functionality. When using system memory 4 to store graphics data, GMCH 3 uses a virtual memory addressing concept for accessing graphics data. 15 In AGP mode, a 32 MB or 64 MB graphics aperture is defined through which addresses in the physical system memory 4 can be accessed by graphics controller 7a. The graphics aperture appears to graphics controller 7a as a 32 MB or 64 MB contiguous block of linear memory, although the addresses in 20 physical system memory allocated for use by AGP graphics controller 7a are not contiguous. The contiguous block of memory addresses in the graphics aperture permits graphics controller 7a to quickly access large data structures, such as

texture bitmaps (typically 1 KB to 128 KB), as single entities in virtual memory.

Access requests from graphics controller 7a address virtual memory within the aperture range, and then GMCH 3 forwards access requests within the aperture to physical system memory 4. The originally-issued addresses sent from graphics controller 7a are translated within data stream controller 26 using a Graphics Address Remapping Table (GART). The GART is a table that matches virtual memory address in the aperture range with corresponding physical memory addresses. The GART is stored in system memory in a location known to GMCH 3 because its location is stored in a register within GMCH 3. Addresses are mapped from the graphics aperture into system memory in 4 KB pages, and each entry of the GART translates one 4 KB page. Thus, when an access request is received from graphics controller 7a in the graphics aperture, the request is momentarily stalled while the appropriate GART entry is fetched from system memory 4. The address of the access request within the graphics aperture is translated using the fetched translation table entry, and the request is forwarded to the physical address in system memory 4 identified by the fetched GART entry.

To speed up memory access requests from AGP graphics controller 7a to system memory 4, GMCH 3 provides a GART entry

cache 28 for locally storing up to four entries from the GART. GART entry cache 28 may also be known as a translation lookaside buffer (TLB). When a GART entry is first retrieved from the GART in system memory 4 to translate a virtual address into a physical address, the entry may be stored in the TLB 28 residing in data stream controller 26. The next time an address request from graphics controller 7a needs to use the same GART entry, the entry can be retrieved from the local TLB 28 rather than from the GART in distant system memory 4. Since GART entries may be stored in TLB 28 and each GART entry provides access to a 4KB page of memory addresses, up to 16 KB of access requests from graphics controller 7a may be translated using the GART entries stored locally in TLB 28 before a new GART entry must be retrieved from system memory 4. If data stream controller 26 needs to use a GART entry that is not stored locally in TLB 28, the necessary entry may be retrieved from system memory 4 and then stored in TLB 28 for future use, thereby replacing an entry previously stored in TLB 28.

Referring again to Figure 2, in Gfx mode, the internal graphics engines 27 of GMCH 3 define a 64 MB logical address space through which addresses in the physical system memory 4 or AIMM can be accessed by internal graphics engines 27. The logical address space appears to graphics controller 7a as a

32 MB or 64 MB contiguous block of linear memory, although the  
addresses in physical system memory 4 or AIMM allocated for  
use by internal graphics engines 27 are not contiguous. Like  
the graphics aperture used in AGP mode, the contiguous block  
of memory addresses in the logical address space permits  
internal graphics engines 27 to access large data structures  
quickly as single entities in virtual memory.

Access requests from internal graphics engines 27 are  
translated within data stream controller 26 using a Graphics  
Translation Table (GTT), which is stored in system memory in a  
location stored by GMCH 3 in a register within GMCH 3.  
Addresses within the logical address space are mapped into  
system memory or AIMM in 4KB pages, and each entry of the GTT  
translates one 4KB page. GTT entries additionally determine  
whether access requests are mapped to system memory 4 or AIMM  
memory, if an AIMM card is present. The same TLB 28 used in  
GMCH 3 to cache GART entries may be used to store up to four  
entries locally from the GTT in order to speed access to  
physical memory. Since the number of GART entries or GTT  
entries that may be stored in TLB 28 is limited by the  
physical die area size of the TLB, using the same TLB to store  
GART entries in AGP mode and to store GTT entries in Gfx mode  
effectively doubles the number of GART or GTT entries that may  
be stored in TLB compared to the number that could be stored

if separate TLBs were used for GART and GTT entries.

Additionally, using the same TLB to store GART entries in AGP mode and to store GTT entries in Gfx mode simplifies the internal logic of GMCH 3 because a single logic serves both functions of the TLB.

Other embodiments are within the scope of the following claims.

006650-449250

What is claimed is:

1           1.    A memory controller hub comprising:  
2           a graphics subsystem adapted to perform graphics  
3           operations on data; and  
4           a cache adapted to store addresses of locations in  
5           physical memory available to the graphics subsystem for  
6           storing graphics data and available to a graphics controller  
7           coupled to the memory controller hub to store graphics data.

1           2.    The memory controller hub of claim 1 further  
2           including a dedicated bus interface coupling the graphics  
3           controller to the memory controller hub.

1           3.    The memory controller hub of claim 2 wherein the  
2           dedicated bus interface includes an accelerated graphics port  
3           (AGP) .

1           4.    The memory controller hub of claim 1 configured to  
2           provide a block of linear, virtual memory addresses for use by  
3           the graphics subsystem, wherein the cache is adapted to store  
4           addresses of locations in physical memory that correspond to  
5           addresses within the block of linear, virtual memory  
6           addresses.

1           5.    The memory controller hub of claim 1 configured to  
2    provide a block of linear, virtual memory addresses for use by  
3    the graphics controller, wherein the cache is adapted to store  
4    addresses of locations in physical memory that correspond to  
5    addresses within the block of linear, virtual memory  
6    addresses.

1           6.    The memory controller hub of claim 1 configured to  
2    provide a first block of linear, virtual memory addresses for  
3    use by the graphics controller and adapted to provide a second  
4    block of linear, virtual memory addresses for use by the  
5    graphics subsystem, wherein the cache is adapted to store  
6    addresses of locations in physical memory that correspond to  
7    addresses within the first block of linear, virtual memory  
8    addresses and to store addresses of locations in physical  
9    memory that correspond to addresses within the second block of  
10   linear, virtual memory addresses.

1           7.    A computer system comprising:  
2           a CPU;  
3           a display device;  
4           a system memory adapted to store video data and non-video  
5   data; and

6 a memory controller hub coupled to the CPU and coupled to  
7 the system memory, the memory controller hub comprising:

8 a graphics subsystem configured to perform graphics  
9 operations on graphics data; and

10 a cache adapted to store addresses of locations in  
11 physical memory available to the graphics subsystem for  
12 storing graphics data and that available to a graphics  
13 controller coupled to the memory controller hub to store  
14 graphics data.

1 8. The computer system of claim 7 further including a  
2 dedicated bus interface coupling the graphics controller to  
3 the memory controller hub.

1 9. The computer system of claim 8 wherein the dedicated  
2 bus interface includes an accelerated graphics port (AGP).

1 10. The computer system of claim 7 wherein the memory  
2 controller hub is configured to provide a block of linear,  
3 virtual memory addresses for use by the graphics subsystem;  
4 and

5 wherein the cache is adapted to store addresses of  
6 locations in physical memory that correspond to addresses  
7 within the block of linear, virtual memory addresses.



1           11. The computer system of claim 7 wherein the memory  
2 controller hub is configured to provide a block of linear,  
3 virtual memory addresses for use by the graphics controller;  
4 and

5           wherein the cache is adapted to store addresses of  
6 locations in physical memory that correspond to addresses  
7 within the block of linear, virtual memory addresses.

1           12. The computer system of claim 7 wherein the memory  
2 controller hub is configured to provide a first block of  
3 linear, virtual memory addresses for use by the graphics  
4 controller and is adapted to provide a second block of linear,  
5 virtual memory addresses for use by the graphics subsystem;  
6 and

7           wherein the cache is adapted to store addresses of  
8 locations in physical memory that correspond to addresses  
9 within the first block of linear, virtual memory addresses and  
10 is adapted to store addresses of locations in physical memory  
11 that correspond to addresses within the second block of  
12 linear, virtual memory addresses.

1           13. A method of storing addresses of locations in  
2 physical in a memory controller hub cache wherein the  
3 locations in physical memory are available to either a

4 graphics controller coupled to the memory controller hub or  
5 are available to a graphics subsystem of the memory controller  
6 hub.

1 14. The method of claim 13 further comprising:  
2 providing a block of linear, virtual memory addresses the  
3 memory controller hub for use by the graphics subsystem; and  
4 storing in the cache addresses of locations in physical  
5 memory that correspond to addresses within the block of  
6 linear, virtual memory addresses.

1 15. The method of claim 13 further comprising:  
2 providing a block of linear, virtual memory addresses in  
3 the memory controller hub for use by the graphics controller;  
4 and  
5 storing in the cache addresses of locations in physical  
6 memory that correspond to addresses within the block of  
7 linear, virtual memory addresses.

1 16. The method of claim 13 further comprising:  
2 providing a block of linear, virtual memory address the  
3 memory controller hub for use by the graphics controller, and  
4 storing in the cache addresses of locations in physical memory  
5 that correspond to addresses within the block of linear,

6 virtual memory addresses; or  
7 providing a block of linear, virtual memory address the  
8 memory controller hub for use by the graphics subsystem, and  
9 storing in the cache addresses of locations in physical memory  
10 that correspond to addresses within the block of linear,  
11 virtual memory addresses.

00660-4492960

A memory controller hub includes a graphics subsystem adapted to perform graphics operations on data and a cache adapted to store of locations in physical memory available to the graphics subsystem for storing graphics data and available to a graphics controller coupled to the memory controller hub to store graphics data.

- 17 -

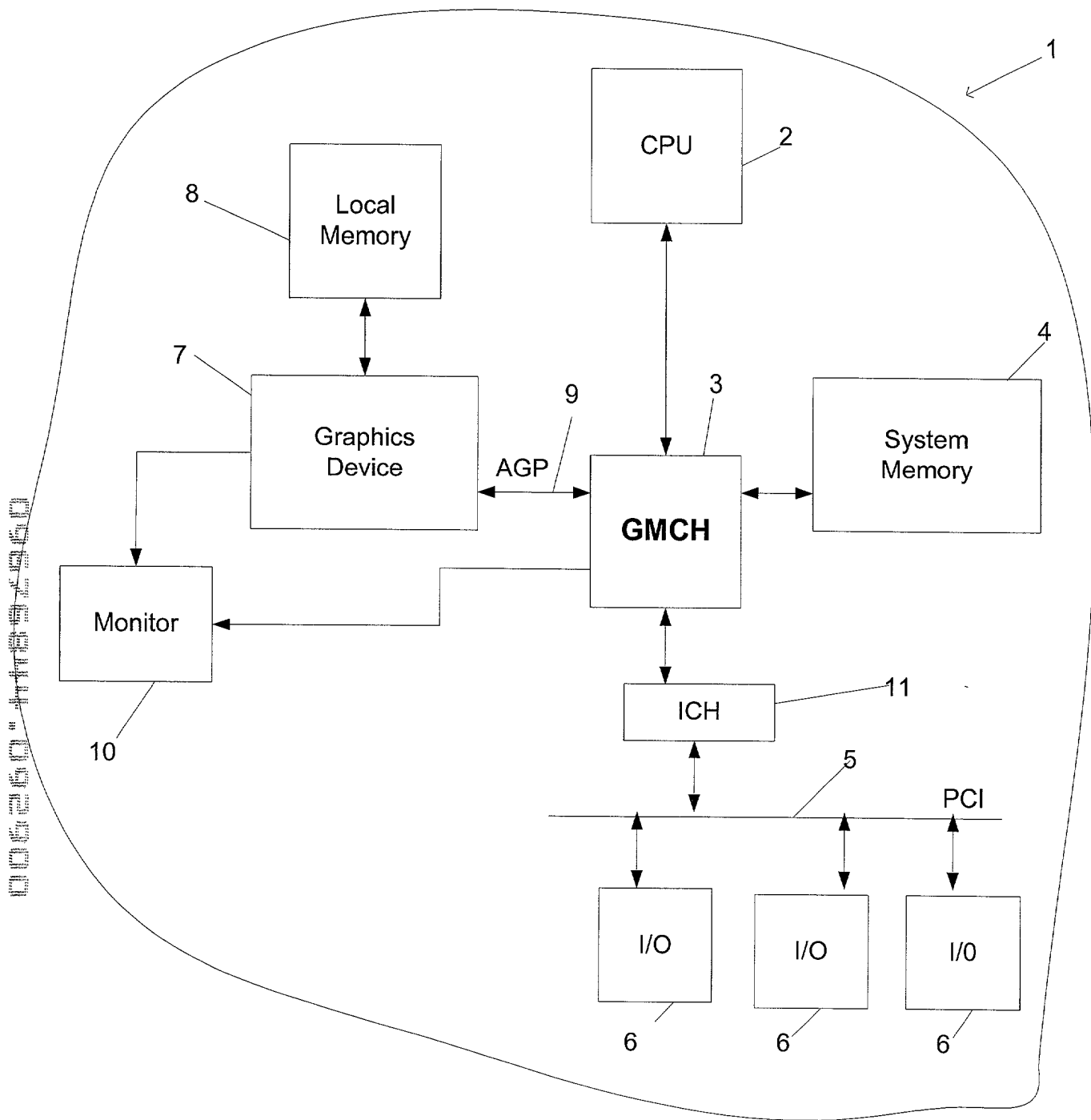


Figure 1

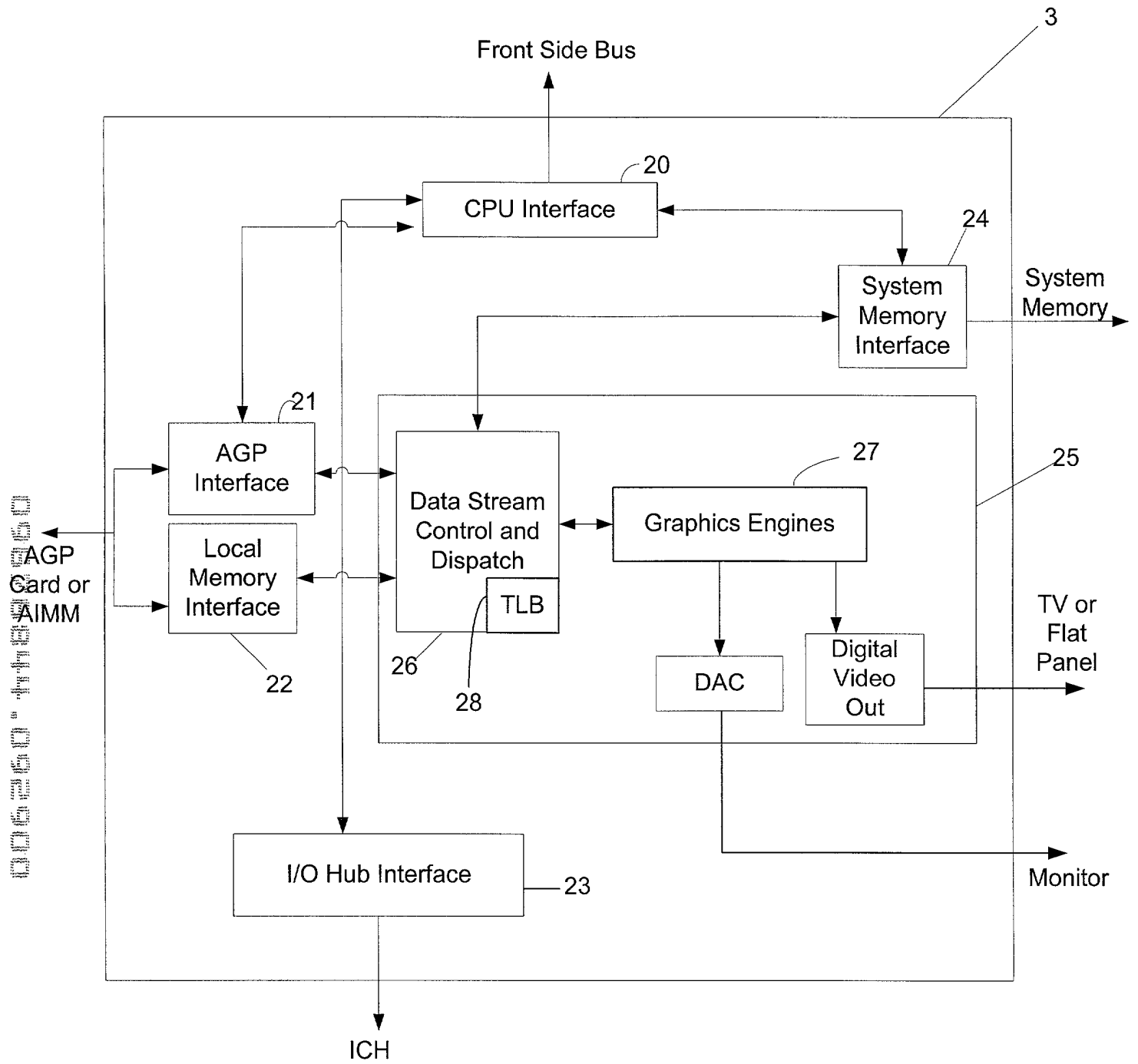


Figure 2

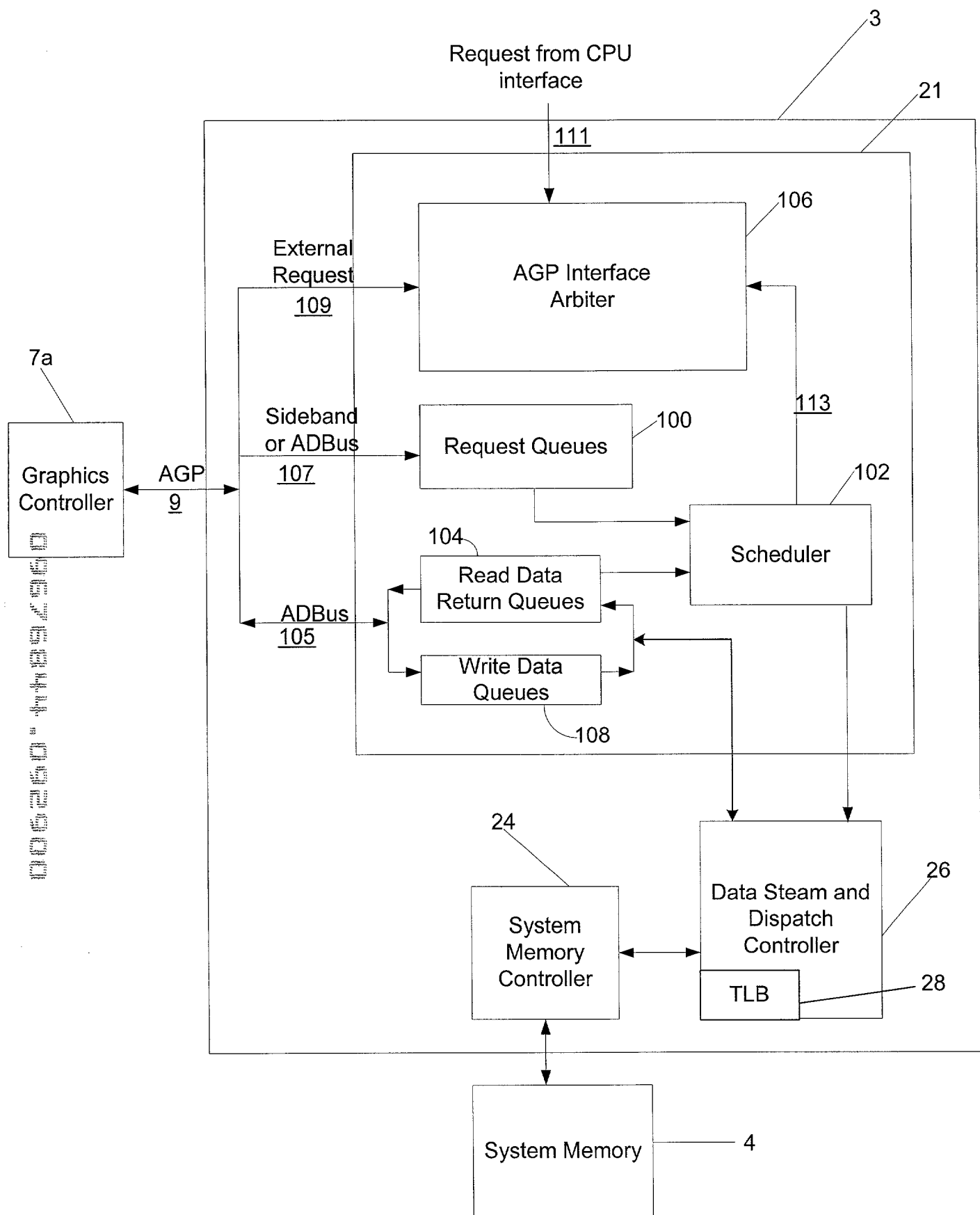


Figure 3

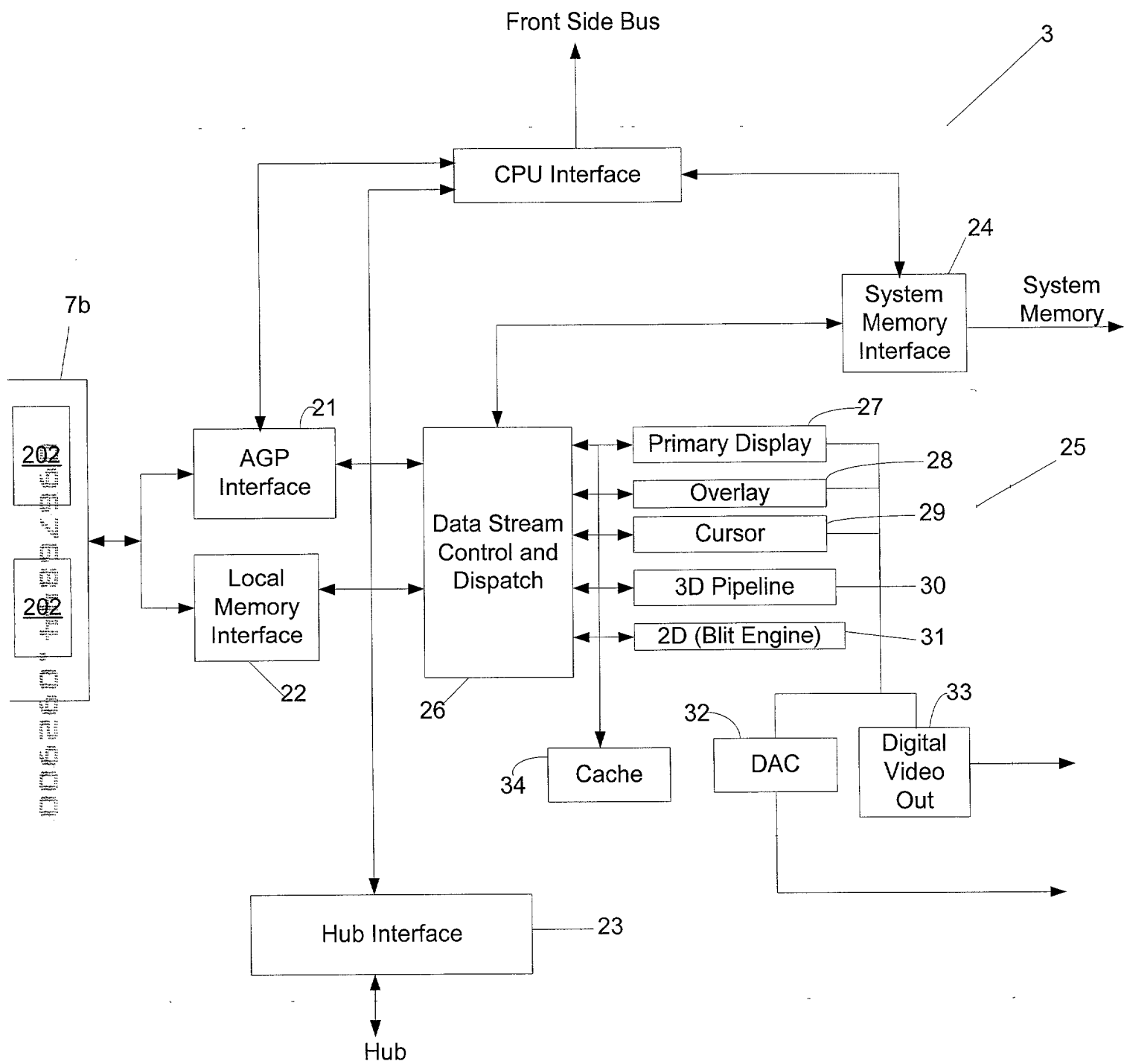


Figure 4